In Situ Eddy Analysis in a High-Resolution Ocean Climate Model

Jonathan Woodring, Mark Petersen, Andre Schmeißer, John Patchett, James Ahrens, Hans Hagen

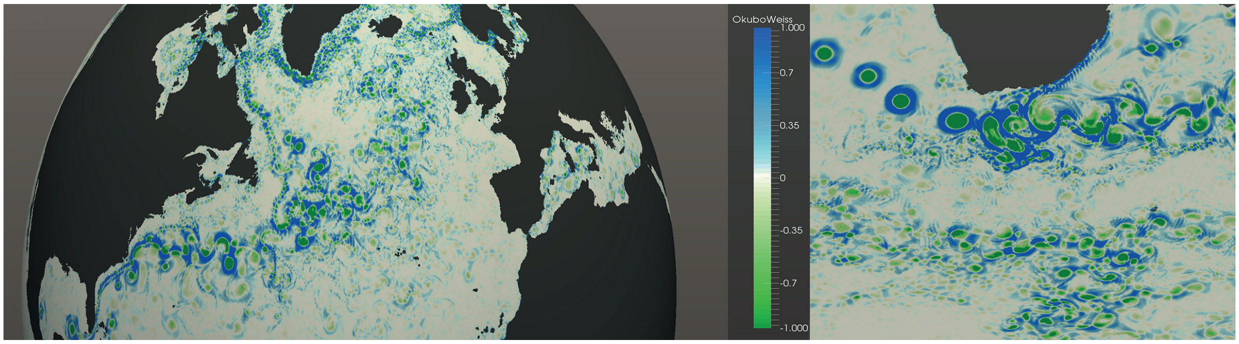

Fig. 1. Visualization of the Okubo-Weiss field from the Gulf Stream (left) and the Agulhas Current (right). These were created with *in situ* eddy analysis in the MPAS-Ocean (*Model for Prediction Across Scales – Ocean*) model for a 15km grid cell resolution simulation. Green regions indicate strong rotation, while blue regions indicate strong shear. The boundary between the regions define an eddy.

Abstract— An eddy is a feature associated with a rotating body of fluid, surrounded by a ring of shearing fluid. In the ocean, eddies are 10 to 150 km in diameter, are spawned by boundary currents and baroclinic instabilities, may live for hundreds of days, and travel for hundreds of kilometers. Eddies are important in climate studies because they transport heat, salt, and nutrients through the world's oceans and are vessels of biological productivity. The study of eddies in global ocean-climate models requires large-scale, high-resolution simulations. This poses a problem for feasible (timely) eddy analysis, as ocean simulations generate massive amounts of data, causing a bottleneck for traditional analysis workflows. To enable eddy studies, we have developed an *in situ* workflow for the quantitative and qualitative analysis of MPAS-Ocean, a high-resolution ocean climate model, in collaboration with the ocean model research and development process. Planned eddy analysis at high spatial and temporal resolutions will not be possible with a post-processing workflow due to various constraints, such as storage size and I/O time, but the *in situ* workflow enables it and scales well to ten-thousand processing elements.

Index Terms—In situ analysis, online analysis, mesoscale eddies, ocean modeling, climate modeling, simulation, feature extraction, feature analysis, high performance computing, supercomputing, software engineering, collaborative development, revision control.

1 Introduction

A traditional, post-processing analysis workflow to produce an eddy census per simulated day is not feasible for large-scale, high-fidelity ocean-climate simulations due to the end-to-end time involved and file storage sizes. In response, we provide an alternative workflow that enables eddy analysis for this ocean model use case, and highlight:

1. The development of an *in situ* eddy census workflow that provides results directly from a simulation run.
2. Implementation of scalable Okubo-Weiss [48] and connected components for generating eddy censuses in MPAS-Ocean (*Model for Prediction Across Scales-Ocean*).
3. New ocean-climate science results that verify the generation of eddies in high-resolution ocean models.
4. Performance results of our *in situ* eddy census compared to the climate scientists' post-processing workflow.
5. Description of our analysis development process, which is peer-reviewed due to being part of the simulation model.

2 Motivation

Computational models are an important tool in climate science to assess the risk of greenhouse gas emissions to future generations [38]. The ocean plays an important role in climate change, including carbon and heat uptake at the sea surface, fresh water fluxes from melting ice caps, and changing currents. The analysis of ocean model data is critically important in climate change research.

In simulations of the earth's oceans, the variables of velocity, temperature, salinity, and mass (the prognostic variables) are computed at every time step using discretized forms of the conservation equations. These prognostic variables comprise the *state* of the system, fully characterizing the ocean model at a particular time. A suite of analysis computations are conducted on the ocean state to gain scientific insights from ocean simulations.

Eddy analysis is one of a large number of analysis procedures applied to the ocean state [11]. Eddies are ubiquitous features in the world's oceans. Many are created by the shear of boundary jets like the Gulf Stream and Agulhas retroflection (Figure 1), where the currents meander and pinch off eddies on each side. Others spawn due to baroclinic instability, in regions where pressure and density gradients are not aligned. The Southern Ocean, which surrounds Antarctica (Figure 2), contains the largest concentration of eddies in the world [28]. This is because waters are colder in the south than the north, and constant density surfaces (isopycnals) are extremely tilted, leading to a large source of potential energy for the baroclinic instability. Eddy formation occurs spontaneously where this is present, and converts the

- *Jonathan Woodring, Mark Petersen, John Patchett, and James Ahrens are with Los Alamos National Laboratory, USA. Email: woodring@lanl.gov, mpetersen@lanl.gov, patchett@lanl.gov, ahrens@lanl.gov.*
- *Andre Schmeißer and Hans Hagen are with Computer Graphics and HCI Group, University of Kaiserslautern, Germany. Email: schmeisser@itwm.fhg.de, hagen@informatik.uni-kl.de.*

Manuscript received 31 Mar. 2015; accepted 1 Aug. 2015; date of publication xx Aug. 2015; date of current version 25 Oct. 2015. For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2015.2467411, IEEE Transactions on Visualization and Computer Graphics

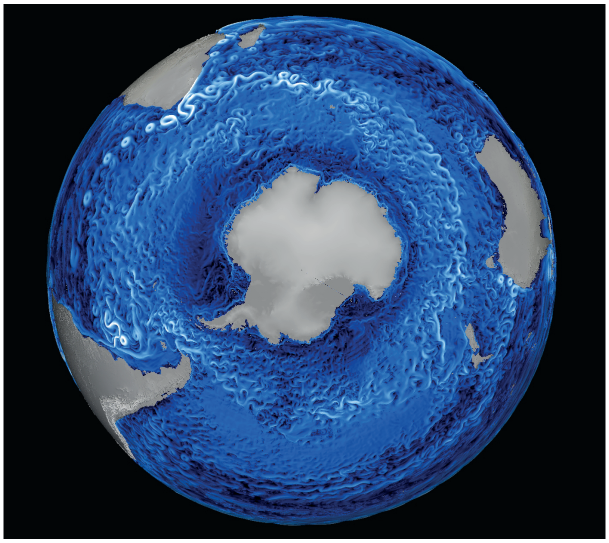

Fig. 2. Visualization of the kinetic energy in MPAS-Ocean. The eddies can be qualitatively seen here, and only appear in the model at high-resolutions (15 km grid cell resolution or finer).

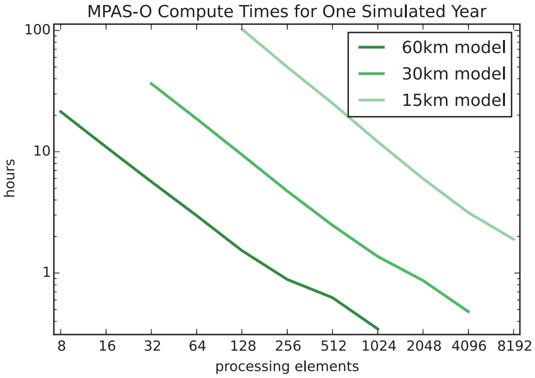

Fig. 3. MPAS-Ocean has nearly linear scalability up to 10,000 processing elements. The 15 km grid cell resolution is necessary for mesoscale eddy formation in the model. The 60 km and 30 km grid resolutions are shown for comparison.

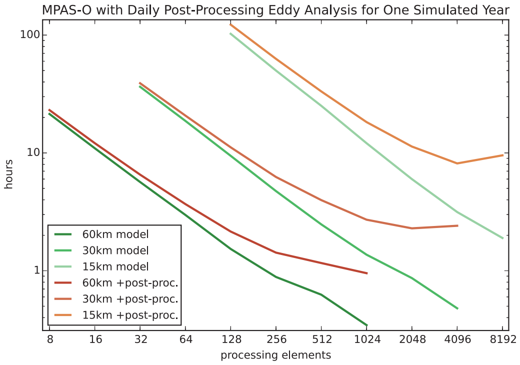

Fig. 4. Adding post-processing time for daily eddy censuses to the total time taken by the simulation. The primary lack of scalability is due to writing the simulation state to storage and then reading it back from storage. This is an *ideal* timing for post-processing, as we have not measured the time that it takes the analyst to copy the files, open them, and run it on a single-core analysis code.

potential energy of tilted isopycnals into kinetic energy of the rotation of eddies [40, 44].

Even though individual eddies occur on scales of 10-150 km, their cumulative effects have large and long-ranging consequences on the earth's climate. In the Southern Ocean, eddies transport heat towards the pole, flattening the isopycnals. The strength of large-scale circulations in the northern hemisphere, such as the Atlantic Meridional Overturning Circulation (AMOC) are sensitive to the eddy dynamics in the Southern Ocean [24, 50]. Winds over the Southern Ocean have become stronger with a warming climate [1], which may lead to long-ranging effects on ocean circulation [14, 20], and eddies play a key role in these dynamics [10]. Eddies have even been evaluated for their relationship to increased fish populations [54].

The diameter of eddies vary from 100 km at 20° latitude, down to 60 km or less at higher latitudes. This is determined by the Rossby radius of deformation, a fundamental length scale associated with the earth's rotation and the vertical density profile [40, 44]. Discretized ocean models need a minimum of four grid cells to resolve an eddy's diameter, so high resolution (15 km or smaller grid cells) is required to study the effects of eddies on global climate. This research, and the associated analysis, is restricted to scalable, high-performance ocean models that run efficiently on large computing platforms.

To perform these studies at scale, the Model for Prediction Across Scales (MPAS) is a climate model framework that supports many components: ocean, atmosphere, sea ice, and land ice. It uses unstructured horizontal meshes based on Voronoi tessellations, which has applications in regional climate modeling [31]. The vertical grid is arbitrary Lagrangian-Eulerian to reduce spurious mixing [29]. The ocean-climate module, MPAS-Ocean is developed at Los Alamos National Laboratory and it has been shown to have nearly linear scalability to the 15 km grid cell resolution needed to resolve eddies, on many thousands of cores, seen in Figure 3. MPAS-Ocean has been validated against analytic solutions and other models in numerous test cases, from idealized to realistic [29]. It has been shown to produce an ocean climate similar to observations, and with performance similar to other ocean models [31].

Climate analysis has been primarily done via *post-processing*, where the ocean state is saved as a *checkpoint restart* file. These files are manually transferred to a local file system, often the analyst's desktop computer, and then, analysis and visualization is conducted with customized code on a single or a few processors. This workflow is unsustainable as climate simulations move to high spatial and temporal resolutions. Ten years ago, a typical checkpoint restart file was 100 MB for a one-degree global grid, while today it is 7 GB for a typical resolution of 15 km grid cells (approximately 1/8th of a degree).

For global eddy analysis, we need to perform daily eddy censuses, which totals to approximately 2.5 TB of storage for a simulated year and 250 TB for 100 years. Figure 4 shows the time taken for one simulated year of daily censuses, using a post-processing workflow on a supercomputer where the analysis scales to the same processor count as the simulation. It shows the total end-to-end time, from starting a simulation to receiving eddy analysis results, where post-processing does not scale beyond a certain number of processing elements and asymptotes to a constant amount of time. It is constant because we have saturated our parallel filesystem capability (I/O bandwidth to storage) and cannot scale any further.

In particular for 15 km grid cells, the eddy analysis itself takes approximately 4 times as long as the simulation (i.e., 8 hours vs. 2 hours for a total of 10 hours) of the simulation at 8k cores. This means it always takes a constant 8 hours on our supercomputers, even with a parallel file system and a parallelized analysis code. We haven't even considered the time that it would take to copy files from archive, the manual labor, or analysis processing on fewer cores (a desktop computer), which is typical in a climate, post-processing workflow. This is because parallel filesystems are a shared resource, and large-scale data sets, especially ones as large as 250 TB, cannot remain stored for very long because they rob resources from other users.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2015.2467411, IEEE Transactions on Visualization and Computer Graphics

Future simulation state files will grow as large as 100 GB and 1.5 TB for 5 km and 1 km global grids, respectively, and the time for analysis will grow even worse. Furthermore, analysis needs to be conducted at a time frequency suitable to the scientific need, such as every simulated day or hour, rather than dictated by fixed I/O, desktop computer, or "an analyst's ability to copy files and open them" constraints. These are the primary arguments for moving to an *in situ* climate analysis workflow, in particular for studying eddies in this paper.

In situ analysis methods are able to take full advantage of mesh partitioning, compute cycles, and parallelism that is tested and efficient in the simulation by directly accessing the state and circumventing reading and writing from the parallel file system. This is assuming we can perform the analysis on an instantaneous or accumulated state over time, as is the case with eddy censuses. A resulting data analysis write, which is created directly from the simulation, may be the same size as a checkpoint restart, but the data products have information targeting the scientific questions at hand, so that expensive I/O time is well-spent towards meeting specific goals. This does reduce the ability to do post-hoc, exploratory analysis, but alternatives are being sought out to get around this limitation [3, 41].

More importantly, moving to *in situ* methods forces a significant change in the climate analysis development process. Post-processing analysis and development created by *individual* scientists tends to have poor version control, documentation, efficiency, and reproducibility due to lack of peer-review and sharing. An *in situ* workflow brings analysis development under the umbrella of the standards of the computational model, such that it has the same process as the simulation code: with planning, testing, documentation, upkeep, and it is available for external peer-review. This team-based development [4] for both model and analysis can be difficult to transition to, due to a cultural shift from a separation of capabilities ("stove-piping") to cross-cutting, but it is drastically critical for long-term climate studies, peer-reviewed science, and policy decisions.

3 Requirements and Background

To meet our scientific goals, we had to break with the traditional post-processing used by climate scientists, to enable large-scale ocean eddy analysis. We used *in situ* processing, which increases the scientists' productivity by producing data artifacts directly from a simulation that are readily analyzable, rather than waiting for post-processing. Additionally, we desired an analysis repository, which would contain the analysis routines and produce simulation executables. Finally, we required that the analysis code base had to be integrated into the simulation development path, such that analyses would be under the same software engineering practices and scientific peer-review, as the model.

3.1 Requirements

There are several desired science outcomes for eddy analysis which drive the design for a scalable analysis workflow.

1. Ocean mesoscale eddies (referred to as *eddies,* hereafter) are important to study to understand for ocean dynamics.

2. To verify and validate ocean models and real-world ocean data, daily eddy censuses must be saved from ocean climate models so they can be statistically compared over space and time.

3. Eddy analysis code must be available for internal and external peer-review and subjected to modern software development processes.

4. The analysis must be *timely* and not burden the analyst. This means that the analysis should not significantly extend the length of time-to-results from the simulation time or require the analyst to wait long periods of time.

This leads to performance and implementation design corollaries of the primary design requirements.

1. Eddies must be generated from the simulation state. Eddies are not a "first-order" variable in a climate simulation, rather a derived quantity that is calculated.

2. Since, we require censuses of eddies over time, they must be extracted as a *feature* (an individual component that can be labeled) from derived data to uniquely identify eddies in space and time.

3. Due to the physical scale of the simulation (15 km grid cells or finer), the analysis needs to have good *strong and weak scaling* properties, matching the simulation performance.

3.2 Background

We build on work performed by Williams et al. [47–49] leveraging the Okubo-Weiss field to identify eddies in the global ocean. This previous work was done in a post-processing paradigm which leads to a limited ability to study eddy censuses over time. Within their series of papers, they developed an eddy analysis workflow. They also describe several other ocean eddy detection methods, in addition to Okubo-Weiss.

Others have also studied methods for visualizing and extracting vortical, eddy-like structures. Shafii et al. [35], Stegmaier et al. [37], and Schneider et al. [34] used the Lambda-2 criterion, which is similar to the Okubo-Weiss and Q-criterion, to build applications for visualizing vortices in turbulent flow applications. Kendall et al. [16, 17] provide climate flow visualization techniques, using a combination of parallel frameworks and geometric feature detection. Silver and Wang [36] and Tzeng et al. [43] provide methods for tracking vortical structures in time-varying volumetric flow fields. Yu et al. [53] performed *in situ* visualizations to study a variety of phenomena in 3-D flow fields. Post et al. [30] provide a survey of flow visualization techniques to provide feature extraction and tracking.

Science-enabling analysis and research has been applied to different application domains, outside of flow and climate visualization, and many are done in post-processing. Large-scale simulations have been moving to *in situ* methods as their primary analysis mechanism [2, 15, 42]. The primary reason is that I/O bandwidth to storage has not kept pace with numerical computing data generation [39], and it is a relatively expensive resource to scale-up compared to compute cycles [18]. While it is possible to achieve performance with post-processing [17, 26], the relative cost to achieve the same level of performance as *in situ* analysis will typically be much higher. Also, it is problematic to store large-scale data for later analysis on parallel filesystems, as data sets are typically flushed to slower archival systems to make room for other users, which decreases post-processing performance. Efforts from Tikhonova et al. [41] and Ahrens et al. [3] have been working towards alleviating the traditional drawback of *in situ* processing, which has been the lack of exploratory, post-hoc analysis.

If we consider scaling back storage use to meet bandwidth costs and constraints, scientific demands are not met. Simulation state changes far more rapidly than a storage subsystem can capture it for later analysis. Therefore, the ratio of simulation state becomes too sparse, and it is not possible to reach scientific analysis goals, such as daily eddy censuses. Due to the relative costs of compute vs. I/O, it is easy to imagine a scenario where the I/O takes longer than the actual computation for a model. Then, it becomes more efficient to restart the simulation to generate necessary data artifacts, rather than save the data for later analysis. We will show that we have reached that inflection point for our large-scale ocean simulations.

Several projects have developed *in situ* tools for circumventing the I/O bottleneck. Fabian et al. [9] enable *in situ* visualization and analysis with the Catalyst library based on ParaView. Likewise, Whitlock et al. [46] provide run-time visualization services with LibSim based on VisIt. Vishwanath et al. [45] and Lofstead et al. [19] provide generalized *in situ* and alternative analysis workflows via their GLEAN and ADIOS frameworks, respectively. Woodring et al. [51] explored zero-copy data structures between simulations and visualization frameworks to reduce the memory footprint of *in situ* analysis. Nouanesengsy et al. [25] developed a generalized framework for triaging large-scale simulation data at the source to produce reduced data products.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2015.2467411, IEEE Transactions on Visualization and Computer Graphics

4 Design and Implementation

In this work, we focused on implementing eddy analysis for a parallel global ocean model, the Model for Prediction Across Scales–Ocean (MPAS-Ocean) version 3 (available at **http://mpas-dev.github.io**) [31], and perform the eddy identification and censusing, described earlier [29]. Eddy fields must be calculated from the simulation state, since they are diagnostic rather than fundamental prognostic variables. The identification of eddies is done in three steps: 1) computation of OW parameter and thresholding of values, 2) finding connected components of thresholded values, and 3) collecting identified eddies and computing statistics such as volume, centroid, and average velocity. The first step is data parallel, while steps two and three require interprocess communication.

4.1 MPAS-Ocean Analysis Framework

Our case-study code, MPAS-Ocean, is developed for global high-resolution climate simulations on supercomputing architectures. The ocean model code utilizes a simulation framework (MPAS), which is shared between other climate models, such as land ice, sea ice, and atmosphere. Both the framework and models use collaborative development using distributed revision control via *git* and *github*. Modifications of the model and framework code must go through an external review process, such that any change must be accepted by different developers that didn't create the new feature.

MPAS-Ocean has been designed from the ground-up to support *in situ* and post-processing analytics, and new analyses must be subjected to the same review process as the simulation model. This code organization of the analysis capabilities has been purposely designed: the goal is that collaborators, who may be unfamiliar with the rest of the code, can be integrated into the development process to add new computations to MPAS-Ocean, with the restriction that any new analysis must maintain the same standards of performance, review, testing, and documentation as the rest of the code. This team-based development model is similar to other large-scale projects, such as VTK, ParaView, and VisIt, and has been anecdotally shown to produce high-quality software (millions of lines of code) from multiple researchers and developers. The key difference is that we are combining model and analytics expertise.

This development model does have the social drawback that it requires cross-cutting and combining of capabilities, rather than a traditional separation and stove-piping. It requires a higher startup cost to development through initial design and requirements planning, along with learning the code base and development strategy of the core MPAS-Ocean team. The benefit and primary reason for this cultural change is that data analytic results will be used and trusted by hundreds of researchers and policy makers for climate studies. Therefore, the analytics need to be subject to the same scientific and development standards as the model. This project development style is advocated by Brooks [4], such that large projects need to draw upon a large set of diverse talents to solve a problem, rather than in isolation.

To accommodate these needs, the analysis computations are separated into individual *analysis members*, each with its own code module, registry of variables, input deck (configuration), and timing intervals for computation and output. To create a new analysis member, a collaborator begins by copying the initial template files. By completing the appropriate sections of the initial code base, the collaborator creates an analysis of the ocean state. At run-time, the ocean state is passed to the analysis member with configuration metadata. All MPAS framework mathematical and I/O operations are available within the new analysis module. The simulation will automatically save any data products created by the analysis member, and this is configurable at run-time by the user.

When the analysis member is complete and tested, it must be reviewed by external and/or core developers and merged using revision control tools (a pull request), at **https://github.com/MPAS-Dev/MPAS-Release**, using an iterative feedback process, which is used by the core team for model and framework changes. This paper is an example of this new collaborative development workflow between model and analytics researchers and developers. One author had no previous experience with MPAS-Ocean, but added the Okubo-Weiss analysis member over a summer. That code was externally reviewed, went through several review iterations by another author, who was also new to the development process, was merged into the master code base, and will be publicly available in the next release of MPAS-Ocean.

MPAS-Ocean allows analysis implementations to be called in two modes: *in situ* at specified frequencies *(forward mode)*, or as a post-processing step using the same analysis code *(analysis mode)*. Although *in situ* analysis is the preferred method, additional analysis is sometimes required after the simulation is complete. Analysis mode allows the user to use a list of checkpoint restart files as input data. This guarantees that the same analysis code is used in both cases, unlike typical post-processing analyses where the code is customized by each researcher. In this paper, all performance and timing data was produced this way, comparing *in situ* and post-processing using forward and analysis modes, respectively.

4.2 Okubo-Weiss

We have chosen to use a method based on thresholds of the Okubo-Weiss parameter, OW, to identify eddies in the ocean. This is one of several methods for extracting an eddy from a flow field [7, 23, 32, 47, 48]. We decided on Okubo-Weiss, rather than other methods, since it scales well in parallel, unlike some of the other methods such as detecting winding streamlines [32]. Okubo-Weiss is the oceanographic term and two-dimensional equivalent of the Q-criterion, that is common in visualizing fluid dynamics simulations [27].

The OW parameter is computed on the velocity field as

$$\mathrm{OW} = S^2 - \omega^2, \tag{1}$$

where S is the horizontal strain and ω the relative vorticity. Both can directly be computed from the gradient tensor of the velocity field. Tensor computations on arbitrary unstructured grids require a weak formulation, where the normal velocities on cell edges are integrated around the horizontal boundary of each cell [33, p. 23]. Only adjacent cells need to be considered to compute this tensor. For cells at the boundaries of computational domains, MPAS provides data about neighboring cells in the form of halos (i.e., ghost cells), since it spatially subdivides the computational domain among processing elements (PEs). Thus, our computation of the velocity gradient tensor is fully data parallel without extra communication, and scales perfectly with increasing number of PEs.

Individual cells are classified as being part of an eddy by thresholding the values as

$$\frac{\mathrm{OW}}{\sigma_{\mathrm{OW}}} \leq t, \tag{2}$$

where σ_{OW} is the pre-computed standard-deviation of the Okubo-Weiss field in the first simulation time step. t is the threshold value, typically -0.2, which can be adjusted by the user prior to the simulation run. This results in a field of booleans that describe which cells belong to eddies, which is processed in the next step.

4.3 Connected Components

Based on the previous step, we compute the connected components of the thresholded OW field. In turn, each connected component is considered to be a distinct eddy. This is a more challenging problem on a distributed-memory architecture compared to shared-memory, as it requires interprocess communication of eddy information across domain boundaries to individual PEs to unify components. Harrison et al. [12] describe an algorithm to efficiently compute connected components in parallel in a general problem setting. They describe a multi-stage Union-Find algorithm, finding connected components, first locally on individual PEs, and then globally across all PEs, together with a spatial partitioning scheme.

We have chosen to implement a simpler algorithm that allows us to reuse the existing infrastructure for interprocess communication in the MPAS framework. Our approach also works in two stages: **1)** use the Union-Find algorithm to compute connected components locally on each PE. This assigns to each cell, which was previously tagged as belonging to an eddy, a globally unique ID. A unique ID per cell can easily be computed as a combination of the local cell ID inside a PE's computational domain and the domain ID, incurring no communication overhead. Union-Find assigns a unique ID out of component cells belonging to a local connected component.

Then in the second stage, we differ from Harrison et al.: **2)** use the MPAS communication infrastructure to iteratively communicate (reduce) the IDs of physically adjacent eddy components (connected across the computational domain boundaries) using halo communication to propagate IDs. Using reduction transitivity, this potentially requires several rounds of halo communication for eddy IDs to converge across PE domains to adjacent components (fully propagate). The theoretical worst case would require a number of communication rounds equal to the number of parallel computation domains. Thus, our solution is asymptotically worse than the algorithm of Harrison et al.

However due to the specific form of the connected components (mostly localized cones that do not weave across the globe), this worst case does not happen in practice. Experimental results have shown that a few rounds are sufficient (typically, between 1 to 4 rounds). The required communication overhead has not shown to be a problem in our implementation, though it is the most significant time in eddy finding. In our studies, the connected component phase takes 85% to 95% of the total eddy census time, and it can be replaced with a better algorithm, such as Harrison et al., should the need arise.

4.4 Eddy Census

With the connected components computed, we finally compute statistics about eddies to create a census. This is first performed on local connected components (partial eddies) and then statistics are globally aggregated. The statistics include the number of cells, the volume, average position and velocity, and horizontal area at selected depths.

The resulting statistics of partial eddies on a each PE is reduced across the global domain. All partial statistics that refer to the same eddy are aggregated by appropriately adding their partial sums of volumes, weighted centroids, etc. This results in a global list of eddies for a single simulation time step that can be used to track eddies across time, which are output to storage. The resulting eddy census data is significantly reduced in size compared to the full simulation data set (by three or more orders of magnitude). This size reduction allows us to track eddies, afterwards, without creating an I/O bottleneck. A future implementation may also track eddies *in situ* to utilize more state information, such as volume overlap between eddies, for more accurate eddy tracking.

5 Ocean-Climate Results

The eddy analysis was applied to five-year MPAS-Ocean global simulations at three global resolutions, each with 60 km, 30 km, and 15 km grid cells. The resolution of eddies is strongly dependent on grid cell size, as eddies typically require at least four grid cells across the diameter, and eddy diameters range from 100 to tens of kilometers. The 60 km, 30 km, and 15 km grids each yielded an average daily eddy count of 8, 180, and 2080. Only the analysis results of the high-resolution grid (15 km) is reported in this section. All eddy census computations were conducted *in situ* daily, and a minimum of 100 grid cells per three-dimensional eddy was required to avoid spurious eddy observations. The output consists of a database of individual eddy characteristics, from which these plots were directly created. The ocean was forced with mean climatological winds [11] and restored to climatological temperature and salinity at the surface. Other model algorithms and physical parameterizations are described in [31].

While numerous studies characterize the eddies in a small region [7, 8, 13, 23, 52], only a handful analyze eddies in global simulations, due to the computing resources required for such an effort (high-resolution over many simulated years). This is the first global eddy characterization with MPAS-Ocean. Statistics may be directly compared with a similar study with the POP model [28] and studies based

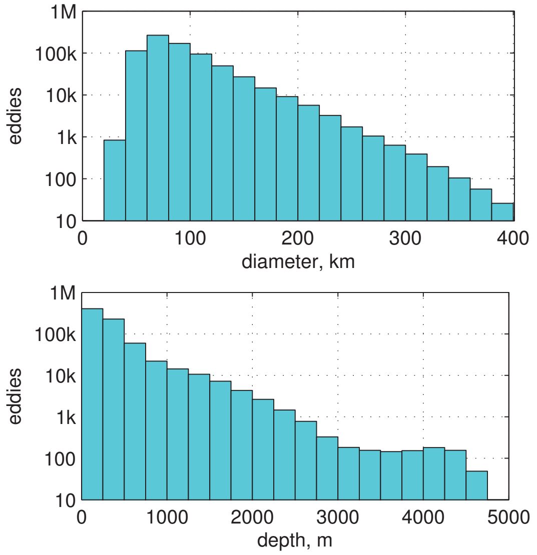

Fig. 5. Distribution of eddies detected by diameter and depth. The majority of eddies are between 50 km and 150 km in diameter, and occur in the top 500 m of the ocean. Results are from a five year MPAS-Ocean global simulation with 15 km grid cells, and the vertical axis is daily eddy observations per year.

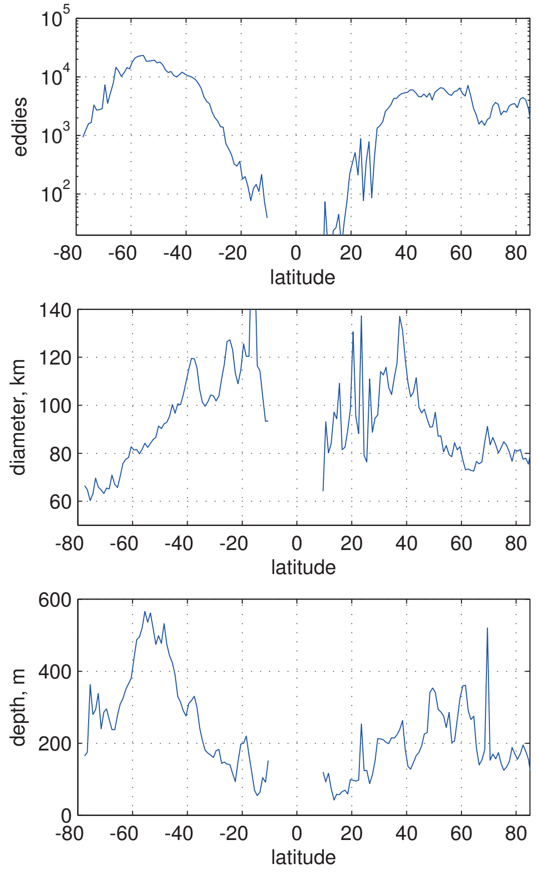

Fig. 6. Eddy statistics as a function of latitude: eddy count, diameter, and depth. Values are averaged over one degree latitude bins over five years. The Southern Ocean, between -40° and -60°, stands out as the location with the largest number of eddies and the deepest eddies. Eddy diameter increases equator-ward due to the Coriolis force, and no eddies exist near the equator.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2015.2467411, IEEE Transactions on Visualization and Computer Graphics

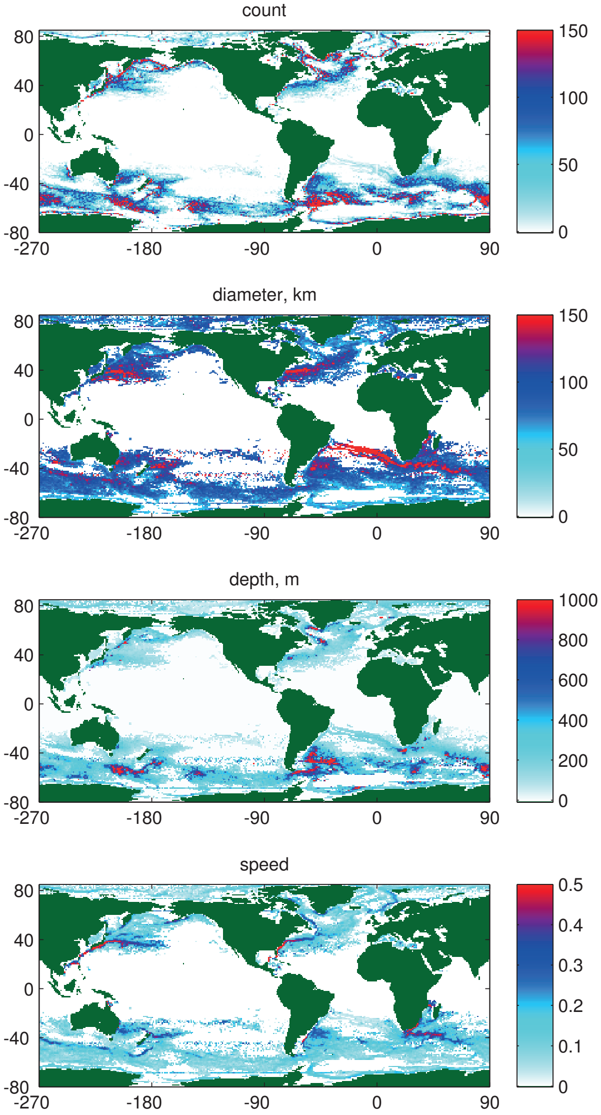

Fig. 7. Eddy statistics averaged by one degree latitude/longitude bins: eddy count, in daily eddy occurrences per year; diameter [km]; depth [m]; and speed [m/s]. Statistics taken over a five year MPAS-Ocean simulation with 15 km grid cells. The Gulf Stream, Kuroshio current (near Japan), and Agulhas current (south of Africa) have the largest and fastest eddies; the Southern Ocean has the largest number of eddies and deepest eddies.

on satellite observations [5, 6]. The advantage of a global study is that the same ocean model and analysis methods are used for all regions, so that global comparisons may be made. Of particular interest are the vertical characteristics of eddies in simulation, as satellite observations only obtain eddy data from the surface and observations with depth are sparse.

In MPAS-Ocean, eddy diameters range from 25 km to 400 km, but the great majority of eddies are between 50 km and 150 km (Figure 5). Eddy diameter is computed as the diameter of a disk with the same area as the maximum horizontal eddy area. Eddy count drops off logarithmically with diameter and this distribution is nearly identical to Figure 11a in the POP study [28]. The minimum eddy diameter is limited by the grid resolution at high latitudes, i.e., 40-60 km for 15 km grid cells. The average depth of the eddy census shows that most eddies occur in the top 500 m, there is a small population below 1000 m, and a negligible number below 3000 m. This matches previous simulations [28] and available observational data [13, 22].

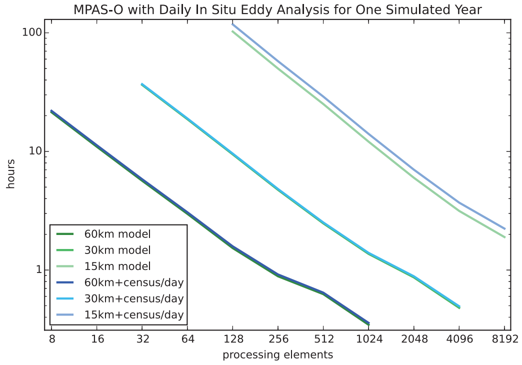

Fig. 8. Comparison of simulation times with and without *in situ* eddy analysis. Our implementation scales linearly with the simulation. The 15 km grid cell resolution is the eddy resolving case and we can see that our eddy census takes relatively longer than 60 km and 30 km cases.

Plotting distributions as a function of latitude, in MPAS-Ocean, reveals patterns in eddy formation (see Figure 6). Eddy diameter is a strong function of latitude, due to the local horizontal component of the Coriolis force. At the equator, the Coriolis force is zero and no eddies appear, while at high latitudes the Coriolis force is strong and eddies are small. Generally, eddy diameter scales with the Rossby radius of deformation, which increases from pole to equator. Statistics between -30° and 30° are noisy due to the small number of eddies there. The baroclinic instability produces a particularly large number of eddies in the Southern Ocean between -40° and -60°. These extend to greater depths than eddies in other regions (compare to Figure 6 in [28]).

Global eddy summary plots provide geographic information on characteristics, which allow oceanographers to correlate eddies with ocean currents and other features (Figure 7). Many eddies are created by the “pinching off” of strong meandering currents, such as the Gulf Stream in the North Atlantic, the Kuroshio Current in the North Pacific, and the Agulhas retroflection south of Africa. These eddies are numerous, large, and fast, but tend to occur in a narrow path along the jet. Eddies also occur in large swaths of the Southern Ocean. The fastest and deepest eddies occur in the Brazil–Malvinas Confluence south of Brazil, and in the Antarctic Circumpolar Current south of Australia (compare to Figure 4 in [28]). Equatorial regions and the center of mid-latitude basins have very few eddies.

The results presented here are important to climate science because of the long-ranging effects of eddies on the ocean. Yet a global eddy census is extremely rare in climate studies because it requires a global, high-resolution ocean model and high-frequency analysis. This work informs oceanographers of the relationship between eddies, currents, and bathymetry, and is a springboard to future studies on eddy transports and their interactions with a changing climate.

6 Performance Results

To create our performance studies, we ran on the “Wolf” supercomputer at Los Alamos National Laboratory. Wolf contains 616 nodes with Intel Xeon E5-2670 processors (16 cores per node) and 64 GB RAM per node. It uses a QLogic QDR InfiniBand network for its interconnect and four scratch filesystems: two Panasas Parallel File Systems and two Lustre Parallel File Systems, in which we used the Panasas in our studies. For I/O, reading, and writing netCDF files (the climate community file standard), MPAS-Ocean uses the PIO library developed by NCAR for the Community Earth System Model (CESM), which is used by many climate models. Internally, PIO uses Parallel netCDF (PnetCDF), which is jointly developed by Northwestern University and Argonne National Laboratory. To perform our *in situ* and post-processing studies, we used MPAS-Ocean in both cases in *forward mode* and *analysis mode*, respectively.

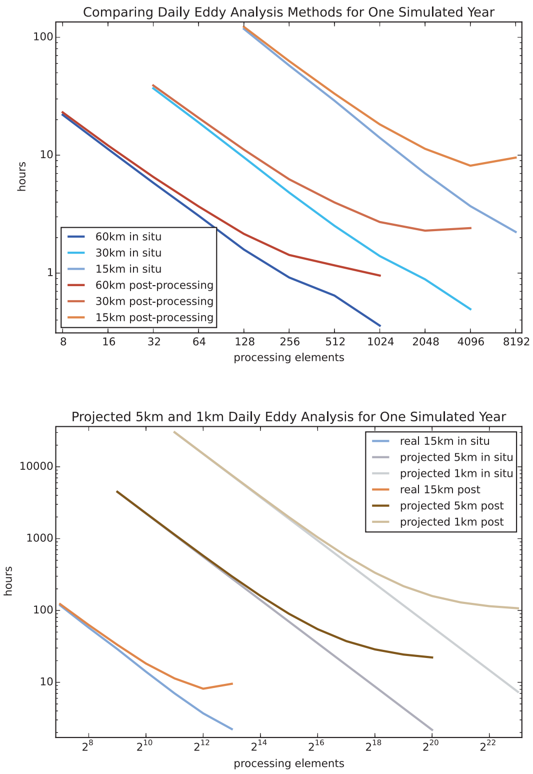

Fig. 9. Comparing total time-to-solution for eddy censuses in our *in situ* implementation vs. our post-processing implementation. Top plot is real, measured times, while bottom plot is projected times at 1-5 km scale compared to real 15 km times.

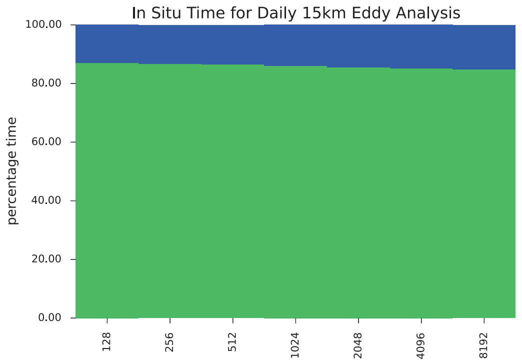

Fig. 10. *In situ* eddy analysis time (blue) relative to simulation time (green). Eddy censuses are nearly constant out to 8k processors. The relative time does increase slightly, as there is room for improvement in our implementation.

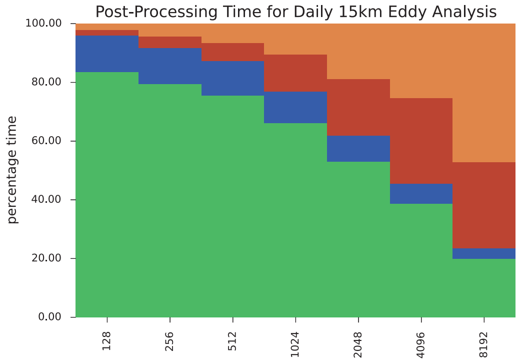

Fig. 11. Percentage of time taken for the post-processing of daily eddy censuses. Green is simulation, blue is eddy census, red is state write, and orange is state read and initialization. As we scale out to 8k processors, we see that I/O takes an increasing amount of time, where I/O takes longer than it does to compute the day.

We show the performance of MPAS-Ocean with daily eddy censuses, over the course of one simulated year, via our *in situ* analysis. This is compared to the time of the simulation without analysis, as seen in Figure 8. The model is run with three different grid sizes, 60 km, 30 km, and 15 km grid cells, which in turn increase the computational model cost by a factor of eight (four times in the horizontal resolution and two times the frequency of time stepping.) With *in situ* eddy analysis, which incorporates the Okubo-Weiss calculation, thresholding, connected components, and census calculation, we achieve the same linear scalability as the simulation code, seen in Figure 10. The difference gap in the 15 km time (Figure 8), with respect to the simulation and compared to 60 km and 30 km times, can be attributed to more eddy components, discussed in more detail in Section 6.1.

In comparing the two workflows, the *in situ* case runs MPAS-Ocean in *forward mode*, which calculates the model and performs eddy censuses directly from simulation state. The post-processing case runs MPAS-Ocean in *forward mode* first (model only), and then, writes its simulation state to parallel file system storage. Next, MPAS-Ocean *analysis mode* reads the state file (checkpoint restart) to calculate the eddy censuses (analysis only, no model). Therefore, the increase in time for post-processing is primarily due to three factors: 1) writing the state, 2) reading the state, and 3) initializing the state prior to analysis. The last factor can be compared to a visualization tool reading in a data set and configuring the data model (for example, reading an HDF5 file and transforming it into a VTK data model).

The time difference between post-processing eddy censuses and our *in situ* solution is drastically different, seen in the top plot of Figure 9. In each case, the post-processing solution doesn't scale and flattens out to a constant time. In the 15 km case, we see an uptick in the total time, as this is due to I/O subsystem, as we have hit the scaling limits of our parallel file system (4k processors, shown by internal testing via our supercomputer production support). Both times, for *in situ* and post-processing, take into consideration the *end-to-end* time to receive a daily eddy census, from running the simulation model and calculating the census. We consider time-to-solution to be the total time to get analysis results, because an analyst relies on images, plots, and statistics to assess a model, not the raw simulation state.

If we project out to the 5 km and 1 km grid cell resolutions, seen in the bottom plot of Figure 9, post-processing analysis becomes an even more unlikely scenario. It would take 100 hours for the post-processing solution to calculate daily eddy censuses. Climate studies investigate long-term scenarios, such as 100 year time frames. This means it would take 10,000 hours, at a minimum, because it cannot scale-out; i.e, over one real-time year (approximately 400 days) to calculate daily eddy censuses for 100 simulated years with a post-processing workflow.

As alluded to previously, we have already reached the point in large-scale simulations where it makes sense to re-run the model with analytics, if a scientist needs a new set of analyses, rather than doing post-processing on the saved state. In Figure 11, at 8k processors, we could re-run the model, with eddy censuses, approximately three to four additional times in the same time that it takes to generate one set of eddy censuses in a post-processing model. (Though, it is faster to save out one, or several, state file(s) to do visualization parameter exploration, then re-run the model once visualization and analysis parameters are picked.) This "re-run the model" trend has been predicted, with other *in situ* research and projects, like Cinema [3].

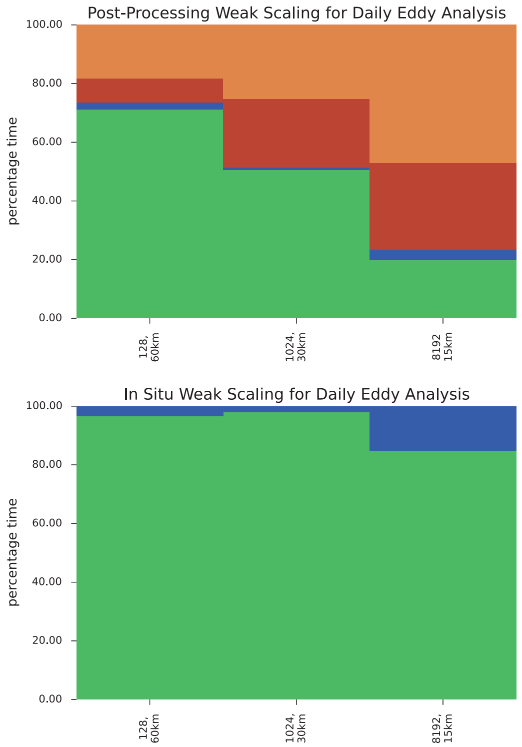

Fig. 12. Relative weak scaling of eddy analysis methods. Percentage time is shown, where green is simulation time, blue is eddy census, red is state write, and orange is state read and initialization. Top plot is post-processing, while the bottom plot is *in situ*. *In situ* takes more relative time at 15 km, because eddy appearance grows faster than the computational complexity of the simulation between 30 km and 15 km.

	128, 60km	1024, 30km	8192, 15km
model + post-proc.	2.2 hours	2.7 hours	9.5 hours
model + *in situ*	1.6 hours	1.4 hours	2.2 hours

Fig. 13. Absolute weak scaling times of model plus eddy finding time. These are the measured times over a simulated year. The increase of time in post-processing is due to I/O, while the increase of time in *in situ* is due to significantly more eddies.

6.1 Weak Scaling

In Figures 12 and 13, we show the weak scaling properties of the two eddy analysis workflows. Between 60 km, 30 km, and 15 km grid cell size, the simulation complexity increases by a factor of 8 (four times in space and twice in time). This is roughly one order of magnitude, and we increase the number of processing elements by 8. As was evident before (top plot in Figure 12, and top row of Figure 13), I/O has poor weak scaling properties as we scale the problem size and processors.

If we look at the *in situ* case (bottom plot in Figure 12, and bottom row of Figure 13), it stands out that there is a jump in the relative eddy census time going from 30 km to 15 km grid cell size. This was also observed in the absolute time plots in Figure 8. While this may seem problematic, it is easily explained. This is due to that eddies “appear” in the model at the 15 km case, while they are not significantly present in the 30 km case. Therefore, the 15 km eddy census calculation has 2 orders of magnitude more work to do, compared to the 30 km eddy census, while the simulation grid complexity has only increased by 1 order of magnitude (a factor of 8).

6.2 Eddy Visualization

We have also studied the scalability of generating images directly from the Okubo-Weiss fields and other variables in the MPAS-Ocean model.

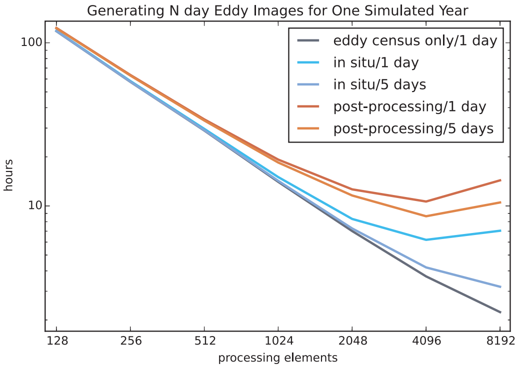

Fig. 14. Generating eddy images with ParaView Catalyst attached to both our *in situ* and post-processing analysis. Images are generated at two frequencies, 5 per day and 5 every 5 days. *In situ* image generation is faster than post-processing at both frequencies. Though, image generation is not scaling as well as we would like it to, and the performance scalability needs to be investigated further.

Figure 14 shows image generation via ParaView Catalyst [9], which has been attached to both MPAS-Ocean *forward mode* and *analysis mode*. In both cases, we generate 5 images at two frequencies, daily images and images every 5 days.

As is expected, *in situ* image generation performs better than post-processing, but the excellent strong scaling stops in this experiment after 2048 cores. We believe this is due to the sort-last compositing used to generate imagery, which has been shown to have flat strong scaling at best on a well-tuned, dedicated and optimized system [21]. We ran our tests on a heavily utilized commodity cluster with an InfiniBand network. This is not problematic for eddy analysis as only a few images are needed for verification and/or debugging. The science need is in the census itself, which does not require high frequency image production.

7 Conclusion

We conclude with a personal testimony on the advantages of *in situ* analysis. One of the authors was involved in a previous global eddy study using post-processing tools. He reports, “The performance numbers show the advantages of smaller file sizes and faster compute times for *in situ* analysis. But in the end, post-processing is such a headache for the analyst, that every new idea seemed like a burden. The *in situ* analysis for this study was so simple by comparison that I was creating final plots just a few minutes after the simulations were done. *In situ* analysis made my work feel easy.”

An *in situ* eddy analysis workflow enables scalable analysis at large spatial grids and temporal frequencies that is not possible with the traditional climate post-processing. This improves assessments of climate change simulations used to inform policy makers. Our case-study indicates that the predicted trend of “re-running” the simulation for new analysis results, due to *in situ* analysis performance scalability over post-processing analysis, is present today. Therefore, scalable simulation codes need to make the transition to a bulk *in situ* analysis workflow, with post-processing methods saved for spot-check verification and debugging.

Secondly, we advocate a cultural shift to model and analysis *co-design* for scientific research and development. This involves integrating the simulation and model development team with the visualization and analysis team, to draw upon both sets of expertise. We believe this has scientific benefits over the post-processing model and separation of capabilities, creating a sum greater than its parts, at the cost of higher startup time and development time through iteration. The benefits for the analysis comes from generating higher quality and reproducible code, due to being subjected to the same development standards as the simulation code: collaborative development and external peer-review via a preliminary requirements and design review, documentation, code review, testing, and public release.

Acknowledgments

We would like to thank Doug Jacobsen, Todd Ringler, and the rest of the MPAS-Ocean core team for their support. The authors would also like to acknowledge the Fraunhofer Institute for Industrial Mathematics ITWM, the Stiftung Rheinland-Pfalz für Innovation, the Los Alamos National Laboratory Laboratory Directed Research and Development Exploratory Research program (20130457ER), the Advanced Simulation and Computing program under the National Nuclear Security Agency, and the Accelerated Climate Modeling for Energy project supported by the U.S. Department of Energy, Office of Science, Office of Biological and Environmental Research.

References

[1] N. J. Abram, R. Mulvaney, F. Vimeux, S. J. Phipps, J. Turner, and M. H. England. Evolution of the Southern Annular Mode during the past millennium. *Nature Climate Change*, 4:564–569, July 2014.

[2] Advanced Scientific Computing Advisory Committee (ASCAC) Subcommittee. Synergistic challenges in data-intensive science and exascale computing – DOE ASCAC data subcommittee report. Technical report, DOE Office of Science, March 2013.

[3] J. Ahrens, S. Jourdain, P. O'Leary, J. Patchett, D. H. Rogers, and M. Petersen. An image-based approach to extreme scale in situ visualization and analysis. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '14, pages 424–434, Piscataway, NJ, USA, 2014. IEEE Press.

[4] F. P. Brooks, Jr. *The Mythical Man-month (Anniversary Ed.)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995.

[5] D. B. Chelton, M. G. Schlax, and R. M. Samelson. Global observations of nonlinear mesoscale eddies. *Progress in Oceanography*, 91:167–216, Oct. 2011.

[6] D. B. Chelton, M. G. Schlax, R. M. Samelson, and R. A. de Szoeke. Global observations of large oceanic eddies. *Geophysical Research Letters*, 34(L15606):5, 2007.

[7] A. M. Doglioli, B. Blanke, S. Speich, and G. Lapeyre. Track coherent structures in a regional ocean model with wavelet analysis: Application to cape basin eddies. *Journal of Geophysical Research*, 112(C05043):12, 2007.

[8] C. Dong, X. Lin, Y. Liu, F. Nencioli, Y. Chao, Y. Guan, D. Chen, T. Dickey, and J. C. McWilliams. Three-dimensional oceanic eddy analysis in the Southern California Bight from a numerical product. *Journal of Geophysical Research (Oceans)*, 117(C16):0, Jan. 2012.

[9] N. Fabian, K. Moreland, D. Thompson, A. Bauer, P. Marion, B. Geveci, M. Rasquin, and K. Jansen. The paraview coprocessing library: A scalable, general purpose in situ visualization library. In *Large Data Analysis and Visualization (LDAV), 2011 IEEE Symposium on*, pages 89–96, Oct 2011.

[10] R. Farneti and T. L. Delworth. The Role of Mesoscale Eddies in the Remote Oceanic Response to Altered Southern Hemisphere Winds. *Journal of Physical Oceanography*, 40:2348–2354, Oct. 2010.

[11] S. M. Griffies, A. Biastoch, C. Böning, F. Bryan, G. Danabasoglu, E. P. Chassignet, M. H. England, R. Gerdes, H. Haak, R. W. Hallberg, W. Hazeleger, J. Jungclaus, W. G. Large, G. Madec, A. Pirani, B. L. Samuels, M. Scheinert, A. S. Gupta, C. A. Severijns, H. L. Simmons, A. M. Treguier, M. Winton, S. Yeager, and J. Yin. Coordinated Ocean-ice Reference Experiments (COREs). *Ocean Modelling*, 26:1–46, 2009.

[12] C. Harrison, H. Childs, and K. P. Gaither. Data-parallel mesh connected components labeling and analysis. In *EuroGraphics Symposium on Parallel Graphics and Visualization (EGPGV)*, pages 131–140, 2012.

[13] S. A. Henson and A. C. Thomas. A census of oceanic anticyclonic eddies in the Gulf of Alaska. *Deep Sea Research Part I: Oceanographic Research Papers*, 55(2):163 – 176, 2008.

[14] A. M. C. Hogg, M. P. Meredith, J. R. Blundell, and C. Wilson. Eddy Heat Flux in the Southern Ocean: Response to Variable Wind Forcing. *Journal of Climate*, 21:608–620, Feb. 2008.

[15] C. Johnson, R. Ross, S. Ahern, J. Ahrens, W. Bethel, K. Ma, M. Papka, J. Rosendale, H. Shen, and J. Thomas. Visualization and knowledge discovery: Report from the doe/ascr workshop on visual analysis and data exploration at extreme scale. *Salt Lake City*, 2007.

[16] W. Kendall, J. Huang, and T. Peterka. Geometric quantification of features in large flow fields. *Computer Graphics and Applications, IEEE*, 32(4):46–54, July 2012.

[17] W. Kendall, J. Wang, M. Allen, T. Peterka, J. Huang, and D. Erickson. Simplified parallel domain traversal. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '11, pages 10:1–10:11, New York, NY, USA, 2011. ACM.

[18] N. Liu, J. Cope, P. Carns, C. Carothers, R. Ross, G. Grider, A. Crume, and C. Maltzahn. On the role of burst buffers in leadership-class storage systems. In *Mass Storage Systems and Technologies (MSST), 2012 IEEE 28th Symposium on*, pages 1–11, April 2012.

[19] J. F. Lofstead, S. Klasky, K. Schwan, N. Podhorszki, and C. Jin. Flexible io and integration for scientific codes through the adaptable io system (adios). In *Proceedings of the 6th International Workshop on Challenges of Large Applications in Distributed Environments*, CLADE '08, pages 15–24, New York, NY, USA, 2008. ACM.

[20] M. P. Meredith, A. C. Naveira Garabato, A. M. Hogg, and R. Farneti. Sensitivity of the Overturning Circulation in the Southern Ocean to Decadal Changes in Wind Forcing. *Journal of Climate*, 25:99–110, Jan. 2012.

[21] K. Moreland, W. Kendall, T. Peterka, and J. Huang. An image compositing solution at scale. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '11, pages 25:1–25:10, New York, NY, USA, 2011. ACM.

[22] R. D. Muench and J. T. Gunn. An arctic ocean cold core eddy. *Journal of Geophysical Research*, 105(C10):23997 – 24006, 2000.

[23] F. Nencioli, C. Dong, T. Dickey, L. Washburn, and J. C. McWilliams. A vector geometry-based eddy detection algorithm and its application to a high-resolution numerical model product and high-frequency radar surface velocities in the Southern California Bight. *Journal of Atmospheric and Oceanic Technology*, 27(3):564–579, 2010.

[24] M. Nikurashin and G. Vallis. A Theory of the Interhemispheric Meridional Overturning Circulation and Associated Stratification. *Journal of Physical Oceanography*, 42:1652–1667, Oct. 2012.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2015.2467411, IEEE Transactions on Visualization and Computer Graphics

[25] B. Nouanesengsy, J. Woodring, J. Patchett, K. Myers, and J. Ahrens. Adr visualization: A generalized framework for ranking large-scale scientific data using analysis-driven refinement. In *Large Data Analysis and Visualization (LDAV), 2014 IEEE 4th Symposium on*, pages 43–50, Nov 2014.

[26] T. Peterka, H. Yu, R. Ross, and K.-L. Ma. Parallel volume rendering on the ibm blue gene/p. In *Proceedings of the 8th Eurographics Conference on Parallel Graphics and Visualization*, EGPGV '08, pages 73–80, Aire-la-Ville, Switzerland, Switzerland, 2008. Eurographics Association.

[27] M. Petersen, K. Julien, and J. Weiss. Vortex cores, strain cells, and filaments in quasigeostrophic turbulence. *Physics of Fluids*, 18(2):026601–+, Feb. 2006.

[28] M. Petersen, S. Williams, M. Maltrud, M. Hecht, and B. Hamann. A three-dimensional eddy census of a high-resolution global ocean simulation. *Journal of Geophysical Research (Oceans)*, 118:1759–1774, Apr. 2013.

[29] M. R. Petersen, D. W. Jacobsen, T. D. Ringler, M. W. Hecht, and M. E. Maltrud. Evaluation of the arbitrary Lagrangian-Eulerian vertical coordinate method in the MPAS-Ocean model. *Ocean Modelling*, 86(0):93–113, 2015.

[30] F. H. Post, B. Vrolijk, H. Hauser, R. S. Laramee, and H. Doleisch. The state of the art in flow visualisation: Feature extraction and tracking. In *Computer Graphics Forum*, volume 22, pages 775–792. Wiley Online Library, 2003.

[31] T. Ringler, M. Petersen, R. Higdon, D. Jacobsen, P. Jones, and M. Maltrud. A multi-resolution approach to global ocean modeling. *Ocean Modelling*, 69(0):211–232, 2013.

[32] I. A. Sadarjoen and F. H. Post. Detection, quantification, and tracking of vortices using streamline geometry. *Computers & Graphics*, 24(3):333 – 341, 2000.

[33] J. Salencon. *Handbook of Continuum Mechanics*. Springer, first edition, 2000.

[34] D. Schneider, A. Wiebel, H. Carr, M. Hlawitschka, and G. Scheuermann. Interactive comparison of scalar fields based on largest contours with applications to flow visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1475–1482, Nov 2008.

[35] S. Shafii, H. Obermaier, R. Linn, E. Koo, M. Hlawitschka, C. Garth, B. Hamann, and K. Joy. Visualization and analysis of vortex-turbine intersections in wind farms. *Visualization and Computer Graphics, IEEE Transactions on*, 19(9):1579–1591, Sept 2013.

[36] D. Silver and X. Wang. Tracking and visualizing turbulent 3d features. *Visualization and Computer Graphics, IEEE Transactions on*, 3(2):129–141, Apr 1997.

[37] S. Stegmaier, U. Rist, and T. Ertl. Opening the can of worms: An exploration tool for vortical flows. In *Visualization, 2005. VIS 05. IEEE*, pages 463–470. IEEE, 2005.

[38] T. F. Stocker, D. Qin, G.-K. Plattner, M. Tignor, S. K. Allen, J. Boschung, A. Nauels, Y. Xia, V. Bex, and P. M. Midgley. Climate change 2013: The physical science basis. Technical report, 2013. 1535 pp.

[39] A. S. Szalay, G. C. Bell, H. H. Huang, A. Terzis, and A. White. Low-power amdahl-balanced blades for data intensive computing. *SIGOPS Oper. Syst. Rev.*, 44(1):71–75, Mar. 2010.

[40] L. D. Talley, G. L. Pickard, W. J. Emery, and J. H. Swift. *Descriptive Physical Oceanography: An Introduction*. Elsevier, sixth edition, 2011. ISBN:978-0-7506-4552-2.

[41] A. Tikhonova, C. Correa, and K.-L. Ma. Visualization by proxy: A novel framework for deferred interaction with volume data. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):1551–1559, 2010.

[42] T. Tu, H. Yu, L. Ramirez-Guzman, J. Bielak, O. Ghattas, K.-L. Ma, and D. R. O'Hallaron. From mesh generation to scientific visualization: An end-to-end approach to parallel supercomputing. In *Proceedings of the 2006 ACM/IEEE Conference on Supercomputing*, SC '06, New York, NY, USA, 2006. ACM.

[43] F.-Y. Tzeng and K.-L. Ma. Intelligent feature extraction and tracking for visualizing large-scale 4d flow simulations. In *Proceedings of the 2005 ACM/IEEE conference on Supercomputing*, page 6. IEEE Computer Society, 2005.

[44] G. K. Vallis. *Atmospheric and Oceanic Fluid Dynamics; Fundamentals and Large-scale Circulation*. Cambridge University Press, first edition, 2006.

[45] V. Vishwanath, M. Hereld, and M. Papka. Toward simulation-time data analysis and i/o acceleration on leadership-class systems. In *Large Data Analysis and Visualization (LDAV), 2011 IEEE Symposium on*, pages 9–14, Oct 2011.

[46] B. Whitlock, J. M. Favre, and J. S. Meredith. Parallel in situ coupling of simulation with a fully featured visualization system. In *Proceedings of the 11th Eurographics Conference on Parallel Graphics and Visualization*, EGPGV '11, pages 101–109, Aire-la-Ville, Switzerland, Switzerland, 2011. Eurographics Association.

[47] S. Williams, M. Hecht, M. Petersen, R. Strelitz, M. Maltrud, J. Ahrens, M. Hlawitschka, and B. Hamann. Visualization and analysis of eddies in a global ocean simulation. *Computer Graphics Forum*, 30(3):991–1000, 2011.

[48] S. Williams, M. Petersen, P.-T. Bremer, M. Hecht, V. Pascucci, J. Ahrens, M. Hlawitschka, and B. Hamann. Adaptive extraction and quantification of geophysical vortices. *IEEE Transactions on Visualization and Computer Graphics*, 17:2088–2095, Dec. 2011.

[49] S. Williams, M. Petersen, M. Hecht, M. Maltrud, J. Patchett, J. Ahrens, and B. Hamann. Interface exchange as an indicator for eddy heat transport. *Computer Graphics Forum*, 31(3):1125–1134, 2012.

[50] C. L. Wolfe and P. Cessi. The Adiabatic Pole-to-Pole Overturning Circulation. *Journal of Physical Oceanography*, 41:1795–1810, Sept. 2011.

[51] J. Woodring, J. Ahrens, T. J. Tautges, T. Peterka, V. Vishwanath, and B. Geveci. On-demand unstructured mesh translation for reducing memory pressure during in situ analysis. In *Proceedings of the 8th International Workshop on Ultrascale Visualization*, page 3. ACM, 2013.

[52] P. Xiu, F. Chai, L. Shi, H. Xue, and Y. Chao. A census of eddy activities in the South China Sea during 1993-2007. *Journal of Geophysical Research*, 115(C03012):564–579, 2010.

[53] H. Yu, C. Wang, R. W. Grout, J. H. Chen, and K.-L. Ma. In situ visualization for large-scale combustion simulations. *IEEE Computer Graphics and Applications*, 30(3):45–57, 2010.

[54] M. Zainuddin, K. Saitoh, and S.-I. Saitoh. Albacore (thunnus alalunga) fishing ground in relation to oceanographic conditions in the western north pacific ocean using remotely sensed satellite data. *Fisheries Oceanography*, 17(2):61–73, 2008.